This document is published in:

Journal of Ambient Intelligence and Smart Environments, 2013, 5, (6), pp. 589-604.
Doi: http://www.dx.doi.org/10.3233/AIS-130231

CALoR: Context-Aware and Location Reputation model in AmI environments

Veronica Venturini, Javier Carbo * and Jose M. Molina
Computer Science Dept., Univ. Carlos III of Madrid, Av. Universidad 30, Leganes 28911, Madrid, Spain
E-mail: {veronica.venturini,javier.carbo,josemanuel.molina}@uc3m.es

Abstract. Spatial conditions of observation are considerably important for some services. Existing distance between the requester and provider agents, while interacting, may influence in a significant way the quality of the provided services. In these cases, recommendations and direct evaluation of services have to take into account such distances. The contribution of this paper is the development of a reputation system that takes into account spatial and temporal properties of interactions for ambient intelligence environments. The system was defined as an extension of an already existing *Protégé* ontology for Ambient Intelligence domains: suggested the corresponding equations (inspired in previous works from other authors): validated the proposal with a case of use, we implemented the corresponding behaviors of *JADE* agents: and executed simulations to show how considering distance may improve reputation accuracy in Ambient Intelligence domains.

Keywords: Ubiquitous computing, trust models, autonomous agents

1. Introduction

Mobile technology applications are increasing. People are immersed in an intelligence world, where cell phones become their assistant, but this world is given to distrust. For example, suppose the following situation in an AmI world extracted from [11]: "Mary is at work and she receives a message from her daughter Anne. Anne wants a new expensive game. Mary does not know that if she bought the game, Anne would play it at school. When Mary receives this special offer, her personal agent buys it, because Anne's birthdays came up. Mary and Anne did not know that the web site (which had the offer) had a spyware that took their personal data and preferences to put them on the market." If Anne's and Mary's agents had requested recommendations from third parties, the negotiation would not take place. Another examples could be going to a little museum or eating out when you are a tourist. Trust systems are required to strengthen and facilitate the final development of ambient intelligence applications in the real world.

*Corresponding author.

Software agents running on mobile devices have spatial-temporal properties, which have to be taken into account at any interaction moment, including when they recommend services of a third party agent to the interacting agent. This kind of recommendations produces a reputational image of that third party agent, enriching trust relationships in agent societies. Although conceptual differences between trust and reputation are still confusing in literature, we adopt the definition of reputation from Jøsang [6], extensively used in the research community. We consider this definition particularly clear in order to efficiently distinguish the concepts of reputation and trust: "reputation is what is generally said or believed about a person's or thing's character or standing" while "trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible."

Reputation systems can be designed as centralized or distributed systems. The first case involves an entity who collects all recommendations and then provides the final reputation outcome to any member/agent of the society. On the other hand, distributed systems con-

sider the case of individuals directly bringing their own recommendations to the member/agent who asked for them; where, finally, the requester computes the reputation based on a third opinion. The centralized approach may have scalability problems since the central entity may become a bottleneck for the system. Furthermore people tend to disagree with giving public access to their own recommendation opinions through a central entity (due to privacy issues) [2]. Taking into account both objections, we developed a distributed reputation system for an ambient intelligence environment where individuals directly interact with each other.

Reputation requests take place consequently when requesting agents have a lack of trust in a particular agent (called target agent). In this case, we ask our trusted partner agents (called recommender agents) to provide helpful recommendations about this target agent. Recommendations are expected to be produced from past direct interactions with such target agents. These recommendations intend to avoid future deceptions with any target agent. Therefore, in general cases, reputation systems have to define:

- how recommendations are computed by recommender agents, through an equation that aggregates the results of past direct interactions with a target agent considering temporal properties (relative *freshness* of such interactions). An example of a reputation system that takes into consideration freshness of information is FIRE model [5].
- how reputation of target agent is computed by requesting agents, through an equation that aggregates the received recommendations considering the reputation of recommender agents.
- how reputation of recommender agents is computed by requesting agents, through an equation that aggregates the level of accuracy of received recommendations according to the final result of the corresponding interaction of the requesting agent indicated by the recommender agent.

A particular case takes place in some ambient intelligence domains where quality of provided services depends on visual and hearing conditions. In such a case, the position that a requesting agent has in regards to the location of the target agent influences how the services were perceived by the user in the interaction. In these cases, the closest positions produce more precise interactions since the ability to discern the quality of services increases. Examples of these ambient intelligence domains could be agents recommending attended live-shows such as theaters and operas, visited monuments, and performed surveillance and recognition tasks. In these kind of domains, spatial information has to play a role in the evaluation of services in some way similar to the temporal information. Supposing a tourist that is visiting a city and has interest in a particular monument. He has a short time before returning home, therefore he decide to ask his contacts about recommendations on such a monument. Each recommender, which interacted with the monument in the past, then sends his own point of view. One of them takes into account a past interaction about two years ago. Another recommender considers, for his evaluation, a past interaction with the monument that took place last week. Also, each evaluation was performed in different positions from the monument. Time and distance are then relevant factors to the expected quality of recommendation. Then, the visitant could build his own image of the monument when he receives these recommendations, in order to decide whether or not visiting the monument. Researchers have generally assumed that interactions that took place longer time ago have less influence in reputation computations, and therefore it is necessary to establish a relationship between reputation values and the time elapsed from the moment the interaction took place (its *freshness*). In an analogous way, we suggest that interactions that were produced far away should also have less influence in reputation computations. This factor leads us to a new kind of reputation system that integrates the distance with target agents into the evaluation of interactions. The justification of the localization usage relies on the assumption that services assessments are more precise when the distance is shorter.

We have seen, that only the work of [3] proposes a reputation system based on collaborative geographic information. It is, in a certain way, a precedent to this work since it also recognizes the relevance of spatial information in the computation of reputation. Particularly, it uses the logarithm of the distances between the agents. The problem we observed in literature is that those reputation systems that include freshness information ignore the distance, while the only system that considers the distance (Bishr proposal [3]) ignores freshness information. We then concluded that it is necessary for a reputation system for AmI domains to conjugate both data: interaction freshness and the distance from where the service was provided. This is the core of our contribution. Unfortunately, whatever the domain we consider, it is not known how the dis-

tance affects the evaluation of a service. We believe that in some AmI domains the interacting distance concerns service assessments, but it is very difficult to know in which (mathematical) way distance is influencing the evaluation of services. In fact, this is similar to when we consider how much relevant is freshness of information to evaluate services and recommendations. There is not an objective way to quantify the dependence between the elapsed time or the interaction distance and the evaluation of the service. Therefore, there is not a precise formulation for them: all proposals are blind adhoc approximations to be tested by simulated data, as we have done.

The structure and outline for the rest of this paper is as follows: Section 2 gives an explanation of the multiagent system (MAS) architecture and ontology used; Section 3 summarizes CALOR communication protocol; Section 4 details the steps of the protocol; Section 5 includes the validation through a case of use; while the implementation is shown in Section 6; Section 7 explains the experiments and obtained results; and finally we conclude in Section 8.

2. Agent system architecture and ontology for a Context-Aware and Location Reputation system: CALOR

We propose a reputation system to be used in the agent system architecture described in [9]. It was designed for heterogeneous domains and to support all activities that could be possible in an Ambient Intelligence domain, such us: trust models, negotiation, location, profile management, etc. The architecture consists of the following agents: Providers (PA), Users (UA), Brokers (BA), Locator (LA) and User Management (UMA) (see Fig. 1). Each user agent runs in the corresponding users' mobile device, while each provider has a server running the corresponding agent provider. The other three agents are running in dedicated servers. Each agent is designed as a deliberative agent which has its own roles, services, beliefs, desires and intentions:

- The Locator agent role is location manager; its beliefs are the location of Users agents and spatial-temporal ontological data.
- UMA has three roles: user manager, coalition manager and search process; its beliefs include public information of user profile.

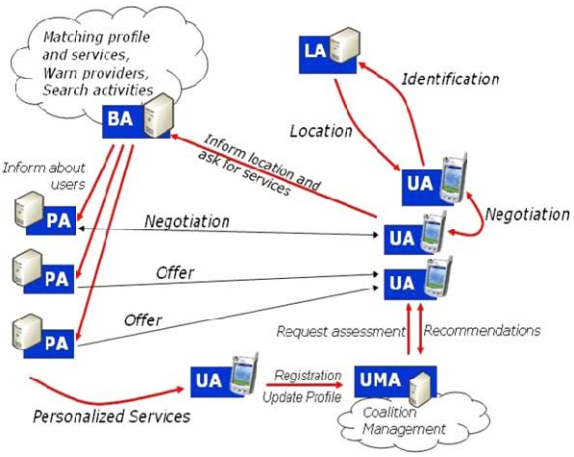

Fig. 1. Overview of the used MAS architecture.

- The roles of Broker agent are: services manager and provider discover; BA believes in the identification of user agents and in the location of Providers agents.
- The role of PA is the provisioning of services and its beliefs are his own contextual information and the information that BA and others PAs share with him.
- User agent roles include: recommendation, negotiation and profile manager roles; Users agents believe in the user profile information.

Notice how the interaction between agents in the MAS is. Firstly, each user has to register into UMA. Next, an User agent has to identify itself with the Locator agent who returns the current position. Locator agent (depending on the system) could be a fusion data agent, if the system works with different location technology like wireless, wimax or ultrawideband as it is in the case of our Lab. Locator agent reasons about the source of spatial-temporal information. User agents according to the user position may be able to request different services from close Provider agents. For these proposes, it has to communicate with Broker agents. This Broker agent is in charge of matching user preferences and the available services in the proximity. Broker agent is responsible for discovering providers to the users. It has also to make the decisions about whether or not to notify the providers about a user. To do it, Broker agent matches a public profile of the user with the provided services by close Providers agents in order to notify providers about the close presence of potentially interested users. Once Provider agent was notified about the specific preferences of users related

to the service that it provides, a negotiation process between the user agent and the provider agent could be initiated.

In addition, prior to this potential negotiation with the provider, an user agent can communicate with other user agents directly to ask for recommendations about such a provider (in this case the former User agent is the requester agent, the latter User agent is the recommender agent and the provider, is the target agent). User Management Agent, plays also a role in these recommendations. For instance, when the requesting agents has no previously trusted agents to request recommendations, UMA would assume the role of finding User agents that could act as potential recommenders. In other case, it could be the requesting agent who contacts directly with recommenders. Additionally, UMA is in charge of handling coalitions of agents with common interests in joining negotiations with provider agents, Provider agent.

In this agent system, users have two profiles:

- a public profile: that is acquired by observing the behavior pattern of the user agents. Although public profile is often associated with the preferences explicitly made public by the user, here we consider public profile the one who was automatically produced through the observation of its acts (movements) by the Locator Agent, LA. On the other hand, the broker agent, BA, is the one who infers user preferences from its movements in a non-intrusive manner (since it is a set of assumptions obtained just from observations). Using this public profile the broker agent will decide which provider agents match the inferred user preferences.
- a private profile: each user agent, UA, has also a private profile that is unknown to any other agent. This private profile is supposed to be an explicit expression of preferences introduced manually by the user. These private preferences are never shared, and they are internally applied by the user/user agent in several decisions: evaluating interactions, considering reputation values and generating recommendations. Due to its source, the private profile will be more specific and accurate, but it requires of the user active participation.

The public profile is then part of the ontology that agents used in our system. It consists then of a set of the interactions of a particular user agent, UA, noted as PastInteractions. Each interaction is defined by a tuple (PA, t, L, ie), where:

- PA providerAgent: instance of a provider agent, PA, it is the PA who interacted with in the past.
- t time: time and date of the interaction.
- L is the location, composed by a pair x-coordinate (noted as Latitude) and y-coordinate (noted as Longitude) of the UA position at the interaction moment. We used them since OpenStreetMap and GoogleMaps use such geographic coordinate position system.
- InteractionEvaluation (ie): it represents the opinion of a requester agent about the interaction quality provided by PA. It is a reputation value calculated by UA over PA. This value is within the interval [1,10].

Additionally, requester agents, UA, have to manage an additional concept to represent the received recommendations. For each interaction tuple of above, we will have several ReceivedRecommendation tuples (RA, PA, ar), each of them is the reputational image that the recommender agent has about a provider agent that shares with the requester agent. In them, RA stands for the UA who acts as the recommender agent, the provider PA and ar stands for the collected information gain sent by the RA about the PA. This value shows how much reputation has the provider PA for the recommender agent RA. It is computed by the RA using past interactions of RA with PA and, after that, it is sent to the UA in response to a recommendation request. Our point is that it should take into account the *freshness* and the distance between PA and RA of each interaction.

Therefore, with these concepts, each UA acting as recommender agent has to compute the Aggregated Recommendation about the provider agent, involving the next additional concepts:

- InteractionFreshness (if): it is computed from the difference between current time and the time of a given interaction. The interact freshness value is between 0 and 1.
- LocationDistance (ld): it is computed from the difference between PA and UA positions at the interaction moment. Location Distance value could be from 0 to thousands.
- InteractionEvaluation (ie): mentioned above.

Finally user agents, UA, acting as requester agents need other two concepts to evaluate recommendations and recommenders:

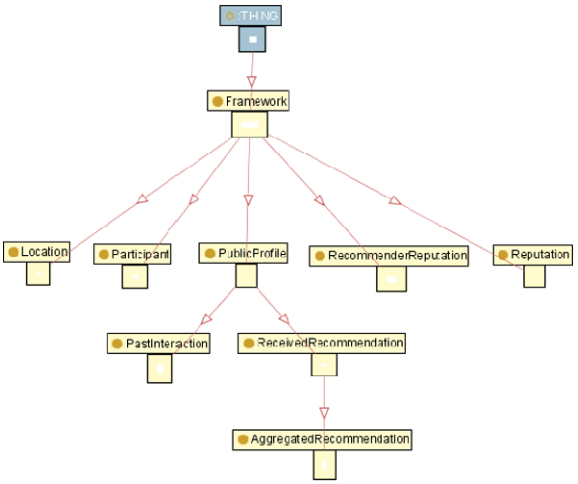

Fig. 2. Ontology extension required to include recommendations.

- RecommenderReputation (rr): it represents how much successful a user agent recommended others PAs in the past for a given requester agent. It is similar to Witness Reputation of FIRE model [5]. This value is computed by the UA who acts as requester agent when it receives a recommendation from the recommender. The rr values is between 1 and 10.
- R Reputation: it is the expected behaviour of the provider in an interval [1,10], it is assigned by the user agent, UA, to the provider agent, PA, in the interaction considering all the previously received recommendations.

Figure 2 summarizes the ontology used by this agents for the reputation system in Ambient Intelligence domains. This ontology is an extension of the ontology exposed in [10].

3. Protocol of a Context-Aware and Location Reputation system: CALOR

Following, we describe the protocol implemented in the provision of a service by a PA. It is summarized in Fig. 3 using Agent Unified Modeling Language (AUML). Each step is detailed in the subsections below. To illustrate it we assume a requester agent UA which asks for recommendations to other user agents (UA^i), acting as recommenders, about provider PA^j. The steps followed then are:

1. Requester agent asks potential recommenders for recommendations: UA asks other UA^i for recommendations about PA^j.

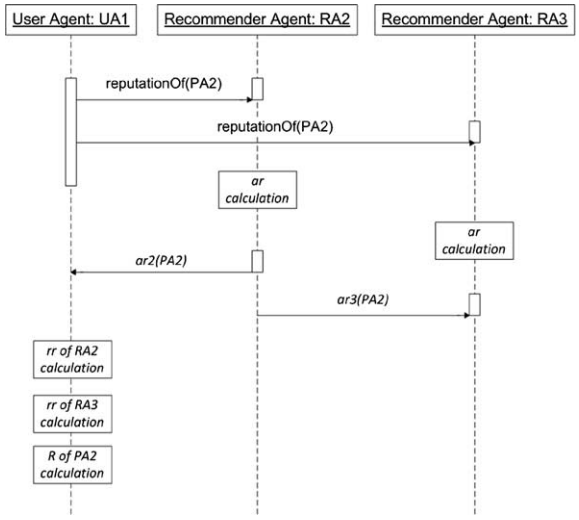

Fig. 3. AUML Interaction diagram corresponding to the recommendation protocol.

2. Recommender agents collect their own past direct interaction with the provider: UA^i searches in their profile for interactions with PA^j.
3. Recommender agents aggregate their own collected past direct interactions into a recommendation to the requester agent: Each corresponding ar slot of a recommender agent UA^i is therefore computed in the recommendation concept sent to the requester agent UA.
4. Requester agent receives and stores ReceivedRecommendation tuples from recommender agents: UA stores a tuple such as (UA^i, PA, ar), for each received recommendation from UA^i.
5. Requester agent computes the reputation of recommender agents: UA computes the RecommenderReputation (rr) of the UA^i, who sent a value in the ar slot of a recommendation concept.
6. Requester agent aggregates recommendations of all UA^i into an reputation value (R) using as weight the just-computed reputation of the recommender (rr) of each UA^i.
7. Based on the obtained reputation value R of provider PA, UA takes a decision about interacting with PA.
8. If the interaction took place, then the user agent (assuming the requester role) would store the (PA, t, L, ie) tuple into PastInteractions table.

Among them, steps 3, 5 and 6 involve computations that will be defined in the next section.

4. Formalization of a Context-Aware and Location Reputation system: CALOR

4.1. Step 3: Recommender agents aggregate the collected own past direct interactions

Since we intend to formulate an equation that takes into account temporal and spatial situations of past direct interactions, we first consider the equations suggested in FIRE model [5] to include freshness of information and the equation of [3] to include spatial distance.

Freshness of information is computed by FIRE model [5] using the equation:

$$f(ie_i) = e^{(-\frac{\Delta t(ie_i)}{\lambda})} \tag{1}$$

Where $\Delta t(ie_i)$ is the difference between the current time and the time when the interaction took place (generating the corresponding interaction evaluation value). While λ is a constant that is set ad hoc depending on the application domain. It is a unit of time expressed in proportional time, it was chosen as a reference in the system.

The exponential function is chosen because we assume that the new ratings are deemed to reflect the target agent's current performance more accurately than old ratings.

On the other hand, in order to take in consideration the physical positioning of an user agent related to a provider agent at the interaction moment, we consider the equation to compute an aggregation of recommendations denominated *nodal degree* from [3] as the main reference:

$$T_m = \sum_{i=1}^{n} \frac{T_i \cdot r(i,m)}{log(c_i)} \quad \text{where } c_i > 1 \tag{2}$$

In this equation the trust T of a provider agent m is T_m, is computed weighting the ratings given by n agents on the system about m ($r(i,m)$) for each interaction i with the trust of recommender agent T_i and the distance (closeness) between them c_i. They chose a $log10$ function to smoothen the sharp variations in the equation implied by using raw distance.

The use of distance logarithm will give more relevance to small differences between two interactions if the distance from the interaction is smaller, while it will give less relevance to the same small differences if the interaction distance of them is bigger. A classic formula of dependency is the function $1/log(x)$. For example, in network density: for a long wire the amount of packet loss is larger than for a short distance (short wire); the logarithm would become almost 1 if the wire was close to the origin. Furthermore, the usage of a logarithm allows us to delimit the distances according to the domain. This is particularly interesting in AmI environments since in them, the domain could be regional, national or international. Therefore, we consider a logarithm-based weight for the distance in reputation computations of Eq. (3). With this computation, we have also found that reputation the final trust values were not heavily penalized by distance influence.

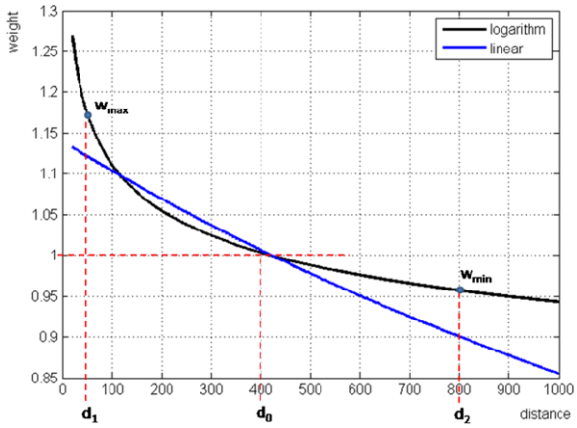

Fig. 4. Logarithm weight function.

$$w = \frac{1}{a + b \cdot log(d)} \tag{3}$$

In Fig. 4 we show the logarithm weight function so we can see how to smoothen the sharp variations in the equation. Black line represents the logarithm functions while blue line represents the linear function. We suppose the following distance configuration: $d_1 = 40$, $d_2 = 800$ and $d_0 = 420$ (d_0 chosen as the central value of the distance). Now we can see how for $d_2 = 800$ the logarithm function falls more slowly than the linear function for the same value. In other hand, for $d_1 = 40$ the difference between both lines is shorter; that is, for shorter distances the w value is higher. From the 800 value, we consider further distance differences less relevant in the computation of reputation. To obtain a and b parameters we require two conditions:

1. The relationship between w_{max} and w_{min}: the min weight has to be a percentage of the max weight;

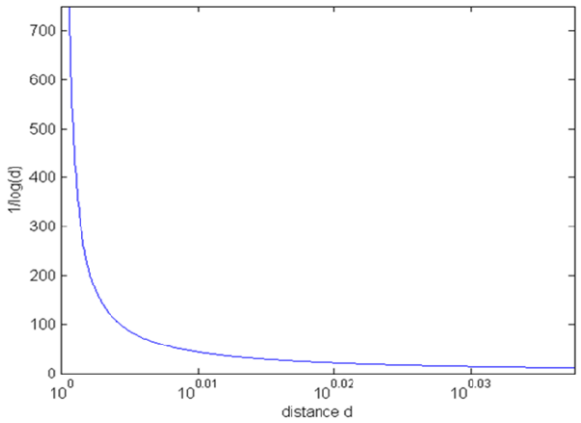

Fig. 5. Logarithm function.

2. A normalization: there is a point where we assume that w is equal to 1.

Both case are shown in the Eqs (4) and (5)

$$w = 1 = \frac{1}{a + b \cdot log(d_0)} \tag{4}$$

$$\frac{w_{max}}{w_{min}} = \frac{1}{a + b \cdot log(d_1)} \cdot \frac{1}{\frac{1}{a + b \cdot log(d_2)}} = \frac{a + b \cdot log(d_2)}{a + b \cdot log(d_1)} \tag{5}$$

Finally, the usage of logarithm function (Fig. 5) allows us to smooth exponential growth of distance.

Given both equations (Eqs (1) and (2)), we propose our own formulation in order to allow recommender agents to aggregate their own interaction evaluations (ie) into a recommendation to be sent to the requester agent. The equation includes both factors: how recently was the interaction and the distance hold between the agents at the interaction moment to weight the ie value. It is a weighted sum of the recommender interactions with the provider, normalized against a reference value. The resulting value (slot ar in the concept recommendation to be sent to the requester agent) intends to be higher when more recent and closer the interaction was. We use a base-10 logarithmic function in a similar way than Bishr did, therefore distance between interacting agents will become less relevant in the computation when they are far away.

$$ar = \frac{a + b \cdot log[\alpha]}{n} \sum_{i=1}^{n} sr_i \tag{6}$$

Where:

- ar (aggregated recommendation) is the final value given by the recommender agent about PA. It is the reputational image that the recommender agent has about the provider agent.
- i is the interaction number between recommender agent and the provider agent.
- a and b are ad hoc parameters needed by the distance function to normalize ar values.
- α: is the average distance in the domain. For instance, if we are in an AmI environment where the distances were in [0..1000] range, α value would be 500.
- n is the total number of interactions recommender agent and the provider agent.
- sr_i (single recommendation): is described just below in the Eq. (7). It represents the relationship between the interaction evaluation of the provider agent and the distance and freshness of the corresponding interaction i.

$$sr_j = \frac{ie_i \cdot f(ie_i)}{a + b \cdot log[c_i(ie_i)]} \; ; \; c_i(ie_i) > 1 \tag{7}$$

Where:

- ie_i is the interaction evaluation value of recommender agent about the provider agent in interaction i.
- $f(ie_i)$: is the freshness of the interaction i as it was computed in FIRE model [5].
- c_i: is the distance (closeness) between the position of recommender agent j and the position of provider agent at the interaction i moment.

4.2. Step 5 requester agent computes the reputation of recommender agents

After each interaction (when a new interaction evaluation, ie, is produced), requester agent has to consider how much successful were the past recommendations. With this intention, it is required to compare for each recommender agent the differences between ar slot of corresponding received recommendations about a provider, and the ie value of the final interaction with that provider. We chose an exponential function, as authors of FIRE model [5] did. The exponential function was chosen given that if the difference between the ar, from the recommender, and the ie, obtained by the target agent, decreases,

Table 1
Provider names, ids and location

Provider	Agent id	Longitude	Latitude
The Museum of High Altitude Archaeology of Salta (MAAM)	PA21	−24.7889941	−65.4111281
Anthropology Museum Juan Martín Leguizamón	PA22	−24.7863838	−65.3975242
Natural Sciences Museum – UNSA	PA23	−24.7961433	−65.4023092

then the recommender will become more trustworthy.

$$rr = 10 * e^{-\sigma} \tag{8}$$

In the case that this exponent were zero (0), the recommender reputation would be 10, that is, recommender agent would be a perfect recommender (the available range is from 0 to 10) in the past, and therefore its future recommendation would have more weight than the others received. This value σ_i stands for:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(\frac{(ar_i - ie_i)^2}{ie_o^2}\right)} \tag{9}$$

With:

- ar_i is the ar slot of a received recommendation about interaction i.
- i: is the interaction number.
- ie_i is the interaction evaluation value of recommender agent about the provider agent in interaction i.
- ie_o is the initial (default) value of ie. It is used to normalize the resulting values through the calculation of the relatives errors to this initial value.
- n is the total number of recommendations received by recommender agent.

4.3. *Step 6 requester agent aggregates recommendations into an indirect reputation value (R)*

Using reputation of recommenders, we can compute the reputation (R) value of provider agents, such as Eq. (7). This R value is obtained through a weighting sum of the received recommendations (their ar slots), where the weights are the corresponding recommender reputation (computed in step 5):

$$R = \frac{\sum_{j}^{m} ar_j \cdot rr_j}{\sum_{j}^{m} rr_j} \tag{10}$$

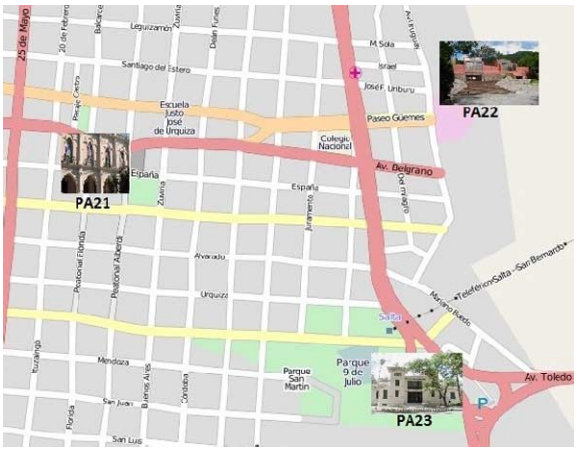

Fig. 6. Map of Salta city with providers.

Where,

- R stands for indirect reputation of provider agent according to received recommendations. It is a value between [1..10].
- ar_j is the ar slot of a received recommendation from recommender agent j.
- rr_j is the reputation of recommender agent j computed in step 5 each time that a new recommendation is received.
- m is the total number of received recommendations for the intended interaction with provider agent.

The success of a reputation system is determined by the similarity of these R values with the real behavior of the provider agent, as we intend to test in advance.

5. Validation of CALOR through an use-case

In this section we show an use-case to illustrate and informally validate the functionality of the architecture, protocol and computations proposed. We assume that several users are visiting the tourist places of Salta city (Argentina), each one with its own address and coordinates. Table 1 shows these places and their corresponding agent ids, names and location.

We can also see in Fig. 6 the map of the city with locations of providers remarked. In order to be an ambi-

Table 2

ie, location and freshness of previous interactions of UA2 with providers

Provider	ie	Longitude	Latitude	Time
PA21	8	−65.4130523	−24.7822804	2010,12,22,7,0,0
PA22	3	−65.4027362	−24.7839892	2010,7,23,7,0,0
PA23	9	−65.4455745	−24.7760736	2011,12,30,7,0,0

Table 3

ie, location and freshness of previous interactions of UA3 with providers

Provider	ie	Longitude	Latitude	Time
PA21	7	−65.4313467	−24.8004947	2010,7,6,7,0,0
PA21	3	−65.4537264	−24.8141731	2007,9,3,7,0,0
PA22	9	−65.4338991	−24.8044136	2012,1,30,7,0,0

ent intelligence domain meaningful for our purposes, we assume that interactions with these tourist places are produced in a more accurate way when user agents visited them from closer locations.

5.1. Initial setup of the use-case

The agents involved in the recommendation process of our use-case are:

- $UA1$: user agent 1, who wishes to know the reputation of provider $PA21$.
- $PA21$: provider agent which reputation has to be computed by $UA1$.
- $UA2$: user agent 2, who acts as recommender of $UA1$ about $PA21$, and who previously had single interactions with $PA21$, $PA22$ and $PA23$.
- $UA3$: user agent 3, who acts as recommender of $UA1$ about $PA21$, and who previously interacted twice with $PA21$ and once with $PA22$.

The Tables 2 and 3 of the initial public profiles of users $UA2$ and $UA3$ shows the details of the interactions mentioned above: interaction evaluation (ie), location where the interaction took place (longitude and latitude) and freshness of those interaction (time):

Additionally we assume that the user agent $UA1$ acting as requester, previously received recommendations from $UA2$ and $UA3$ about $PA22$ and $PA23$ corresponding to the interactions they have had with these providers. Table 4 shows the received recommendations from $UA2$ and $UA3$ about $PA22$ and $PA23$. The results of the interactions of $UA1$ with $PA22$ and $PA23$, that took place after such recommendations, are shown in Table 5 and they form the public profile of $UA1$ (see Fig. 7).

The ar value of the last column of Table 4 is computed using the ie values of both interactions of $UA2$ with $PA22$ (according to the times showed in the previous Table 2) while the other ar values are computed from single ie values.

Table 4

Recommender agent, recommended provider and received recommendation of UA1 with UA2 and UA3

Recommender	Provider	ar
UA2	PA22	0.350227674
UA2	PA23	2.265593464
UA3	PA22	4.152888877
UA3	PA22	4.152888877

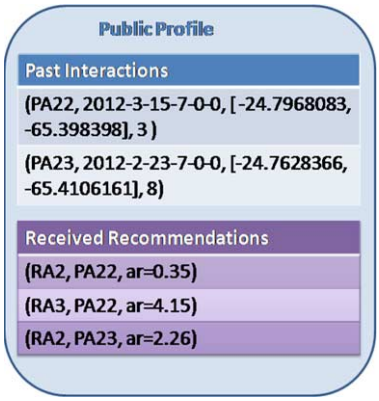

Fig. 7. UA1 public profile.

5.2. Agents executing an instance of the protocol in the use-case

The protocols initiate when user agent $UA1$ sends messages requesting recommendations (see Fig. 8) to $UA2$ and $UA3$ about $PA21$, since both agents interacted previously with $PA21$, and both of them previously recommended $UA1$ about interactions hold with $PA22$ and $PA23$.

Table 5

ie, location and freshness of previous interactions of UA1 with providers

Provider	ie	Longitude	Latitude	Time
PA23	8	−65.4106161	−24.7628366	2012,2,23,7,0,0
PA22	3	−65.398398	−24.7968083	2012,3,15,7,0,0

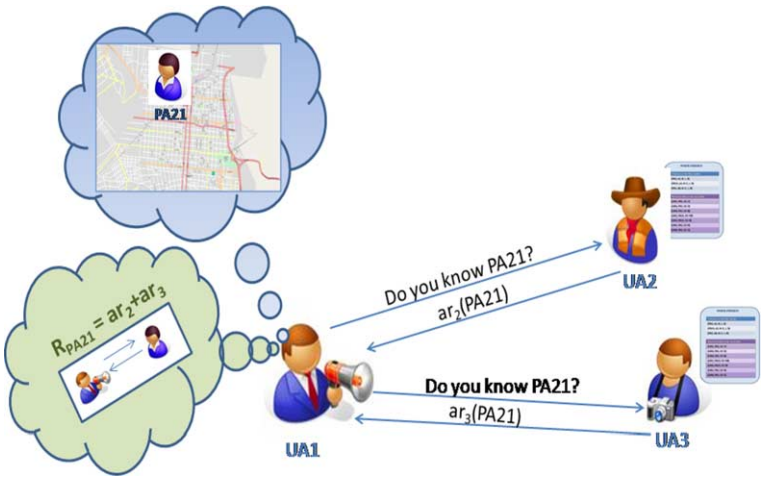

Fig. 8. UA1 call for recommendation.

Each of the agents acting as recommenders ($UA2$ and $UA3$) receives the requesting recommendation message and starts the corresponding internal computation to provide a recommendation (noted as ar) about $PA21$. This internal computation corresponds to the step 3 of the protocol in Eq. (6) (ar value). Where each sr is computed from Eq. (7).

So replacing the corresponding variables with the values of $UA2$ recommending $PA21$:

- $a = -1.142$
- $b = 0.857$
- Interaction: $ie_1 = 8$

Since $UA2$ interacted once with $PA21$, we just have a single sr_j value of 0.94 (and therefore $n = 1$). And next, we replace variables with the corresponding values to obtain ar from:

- $a = -1.142$
- $b = 0.857$
- Weighted interaction: $sr_1 = 0.94$
- $n = 1$
- $\alpha = 400$

Therefore the recommendation that $UA2$ sent to $UA1$ about $PA21$ is an ar valued as: 1.033

Now we repeat the same computation to obtain the ar value sent by $UA3$ to $UA1$ about $PA21$. In this case we have two interactions of $UA3$ with $PA21$:

- $a = -1.142$
- $b = 0.857$
- Interaction 1: $ie_1 = 3$
- Interaction 2: $ie_2 = 7$

We obtain then two sr_j values of 0.0155 and 3.51016 (with $n = 2$). And next, we replace variables with the corresponding values to obtain ar from:

- $a = -1.142$
- $b = 0.857$
- $n = 2$
- $\alpha = 400$ (possible distance at the domain: [0:800])
- Weighted Interaction 1: $sr_1 = 0.0155$
- Weighted Interaction 2: $sr_2 = 3.51016$

Therefore the recommendation that $UA3$ sent to $UA1$ about $PA21$ is an ar valued as: 1.9179

$UA2$ and $UA3$ then answer the requesting recommendation message sent by $UA1$ with the recently computed ar values, in a completely transparent way to the user (whose agent is acting the role of recommender). When $UA1$ receives the recommendations of $UA2$ and $UA3$, it computes the recommender reputation of them from the relative level of success of the previous recommendations of $UA2$ and $UA3$ about $PA22$ and $PA23$. This computation corresponds to the step 5 of the protocol, rr value, Eq. (8). Where σ is computed from Eq. (9).

So replacing the corresponding variables with the values of $UA2$ recommending $PA22$ and $PA23$:

- Recommendation about $PA22$:
  - $ar_1 = 0.35$
  - $ie_1 = 3$
- Recommendation about $PA23$:
  - $ar_2 = 2.26$
  - $ie_2 = 8$
- $ie_o = 3$
- $n = 2$

We obtain a σ value of 4.4704, which provides a recommender reputation (rr) of $UA2$ valued as 0.1144.

In the case of $UA3$ recommending (twice) $PA22$, we had:

- Recommendation 1 about $PA22$
  - $ar_1 = 4.15$
  - $ie_1 = 3$
- Recommendation 2 about $PA22$
  - $ar_2 = 4.15$
  - $ie_2 = 3$
- $ie_o = 3$
- $n = 2$

And then we obtain a σ value of 1.15, which provides a recommender reputation (rr) of $UA3$ valued as 3.16.

Once we have the reputation of both recommenders (rr), we can now aggregate both received recommendations (ar) into the final reputation value R of provider $PA21$. This computation corresponds to the step 6 of the protocol. In this step we compute R from Eq. (10).

So replacing the corresponding variables with the values of rr about $UA2$ and $UA3$, and the values of ar received from $UA2$ and $UA3$ about $PA21$:

- $ar_0 = 1.033$
- $ar_1 = 1.9179$
- $rr_0 = 0.1144$
- $rr_1 = 3.191$
- $m = 2$.

In this way, our case of use would have obtained a reputation R of $PA21$ valued as 1.8872. Which is a reasonable value given the small number of interactions from which the recommendations were computed. But if we increase the number of interactions and recommenders then results will become more relevant, as we can see in the following section.

6. Implementation of a Context-Aware and Location Reputation system: *CALOR*

Agents of *CALOR* system were developed using *JADE* library [1]. We implemented the five types of agents with their behaviors described in [10]. Two additional behaviors were developed by user agents. One of them is *Ask for reputation* behavior: It is the behavior that run the user agent $UA1$ to request services from provider agent $PA21$. The other behavior is *Response Reputation* behavior, which was run by the other user agents: $UA2$, $UA3$ and $UA4$, who supposedly interacted with $PA21$ in the past. They acted here as recommenders. Agents communications were *FIPA* messages that included the mentioned above concepts of a *Protegee* ontology. The simulation started when $UA1$ sent a *Request* FIPA message to each agent of the system, indicating which provider supplies the information that he wants to receive. The other user agents acting as recommenders started computing the value for the collected aggregated recommendation slot (ar). ar value was computed according to the equations described in this paper. Recommender agents then sent an *Inform* message which contains ar value. Next requester agent, $UA1$, decided to interact with the provider $PA21$. Requester agent $UA1$ sent to $PA21$ a *Request* message. Now, they would start the afterwards negotiation process exchanging additional messages. Figure 9 is a screen capture of the running *JADE* agents' communications corresponding to this example.

7. Model evaluation through simulations

Real-world conditions seem to be feasible to simulate [4]. Some other works [7,8] use this kind of validation with simulations since they consider that the reality exceeds their availabilities and capacities. Assuming this point of view, in this section, we show a Montecarlo simulation to run our validation experiments.

7.1. Initial conditions of simulation

Each simulation involved 20 user agents and 5 provider agents. The user agents could ask about rep-

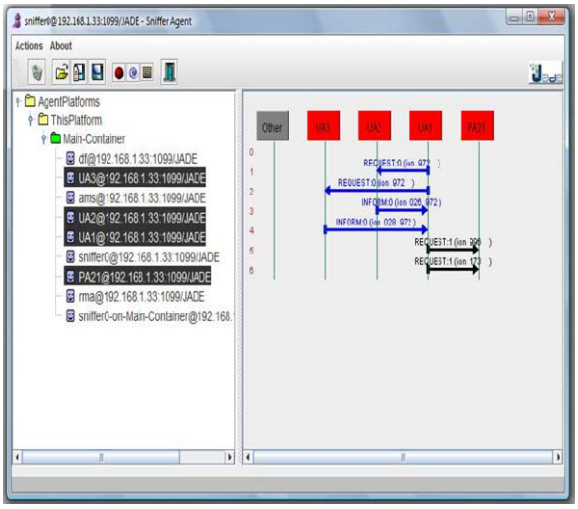

Fig. 9. JADE agents running CALOR system.

utation of any provider. Any user agent could, anytime is requested to, assume a recommender role. Initially, agents have not data over past recommendations, but they have initial information about past own interactions (a table with 50 initial interactions of all agents). This table contains the ie data (interaction evaluation) of user agents UA with provider agents PA. This table was created with the following information:

- UA: a random number obtained from an uniform distribution between [1..20],
- PA: a random number obtained from an uniform distribution between [1..5],
- ie: a random number obtained from a normal distribution, with the mean and variance correlated to the quality of the services offered by provider,
- $freshness$: a random number obtained from gamma distribution with $\alpha = 0.8$ and $\beta = 0.1$ to provide freshness values between [0..1],
- interaction $distance$: a random number obtained from a gamma distribution with $\alpha = 16$ and $\beta = 32$ to provide distances values between [0..800].

We assume that each provider has a inherent quality that determines the quality of the services provided in the interactions ie. This quality is represented by a standard Normal distribution $N(\mu_p, \sigma_{p,c})$, where the mean represents the quality of provided services $\mu_p \epsilon [1..10]$, which dependents on the distance from the service that was provided, while the variance represents the variability of such quality $\sigma_{p,c} \epsilon [0.6..1.6]$. A different value of ie variance would then produce different types of simulations, which will represent new

Table 6
Providers' services quality to provide an interaction evaluation, ie

Provider	ie mean	ie variance
1	3	1.217
2	5.75	1.422
3	8.13	0.831
4	7.17	1.344
5	4.33	0.992

Ambient Intelligence scenarios. For instance we can observe in Table 6 the quality service (mean and standard deviation used in the normal distribution) of 5 providers of our simulation. Then, ie variance values of these normal distribution to produce a particular ie value are high enough to define an scenario of providers with enough instability (high variability) generating services.

While the agents run, the recommendation process evolves. We then generate 50 simulations. Each simulation is a cycle of 15 recommendation request/reply rounds. In each round we select randomly (uniform distribution) an agent user to act as requester, UA, and a provider agent PA. We assume that the chosen agent user is always diligent to ask for recommendations about the chosen provider agent. Therefore, the requester agent, UA, looks at the initial interactions table searching if any other user agent interacted previously with the provider PA. An array of recommenders would be then the result of such search in the initial interactions table. Each potential recommender of this array would then asked for recommendations. Next, the recommender agent would compute the corresponding ar values about provider PA which would be given to the requester agent, UA. With those ar values, the requester agent UA would finally compute the reputation R of the provider PA.

Therefore, in the computation of R, the requester agent has to consider for each recommender agent:

- The time when recommender agent sent his aggregated recommendation value (ar).
- The aggregated recommendation value (ar) (Eq. (6)) which included:
  - The $freshness$ of the interaction between the recommender and the provider
  - The $distance$ between both interacting agents
- The recommender reputation, rr, Eq. (8).

As it was explained for ie variance, different definition of gamma parameters of $freshness$ and $distance$ would then produce different types of experiments,

Table 7
Relative error of predictions based on reputation considering and ignoring distances

Provider	Relative error ignoring	Relative error considering
1	0.196667	−0.03
2	0.373217	0.171304348
3	0.224846	0.045756458
4	0.208926	−0.00223152
5	0.206236	0.047113164

which will represent new Ambient Intelligence scenarios. As an example, we consider initial values of these gamma distribution parameters, to produce an scenario with long distances and recent interactions with $\lambda = 20$ and 0.01 respectively.

7.2. Evaluating distance role

With this initial setup, we aim to analyze how distance influences the final reputation of provider comparing how close is, the reputation R, to the real Service Quality in two scenarios:

- Considering distance between interacting agents
- Ignoring distance between interacting agents. With the propose to compute reputation without distance, we subtract the $distance$ on the Eq. (6), and the resulting equation becomes:

$$ar = \frac{1}{n}\sum_{i=1}^{n} ie_i \cdot f_i(ie_i) \tag{11}$$

Figure 10 shows how close were the reputation values (R), considering and ignoring the distances, to the means of the generated interaction evaluations, ie, for each provider (ar). Table 8 shows the relative error of such predictions considering and ignoring the distances. It shows how reputation (R) value considering distance is closer to the mean of ies, it had an average error of 0.04 while the simulations ignoring distance had an average error of 0.24.

If we just focus on provider $PA2$, we will draw the histogram of the evaluations of interactions (ie) with $PA2$ (drawn in blue in Figs 11 and 12), and then we will see where the average final reputational images (R) of $PA2$ with all the users (drawn in red) along all the simulation. Figure 11 shows the simulation considering the distance and Fig. 12 shows the simulation ignoring the distance. We can observe then that reputation of $PA2$, when including the distance in the computations, has a probability of success about 15%

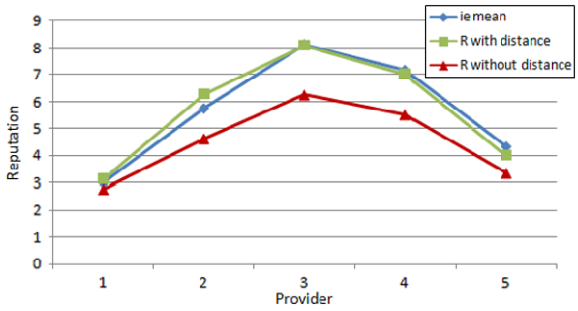

Fig. 10. Reputation of each provider (R) and mean of the interaction evaluations (ie) considering and ignoring distance.

while reputation of $PA2$ when ignoring the distance has a probability about 5%.

These results show that, if the evaluation of interactions depends upon the existing distance between interacting agents, our proposed equations will improve trust models that ignore this factor. Since we do not count on real data to prove the accuracy of our computations we can only show simulations from randomly generated data. Additionally, since a trust model considering distance between agents is an innovative contribution, there are no other similar trust models to compare with, we just can show the inclusion of distance may improve the predictions based on recommendations in domains of Ambient Intelligence.

In the next section, we compare the performance of CALoR with other proposals facing experiments that consider the influence of interacting distance and the real behavior of the providers collected in ie values.

7.3. Comparing CALoR with other reputation systems

In order to evaluate the performance of CALoR system, we compared it with Bishr and Fire reputation systems. The experiment with Fire and Bishr models, consists of 50 simulations with 5 interactions each round. The setup of Bishr adaptation to our experiment is the following: trust t, a value extract from a normal distribution between 1 and 10; rating r, a value of a uniform distribution between 0 and 1; c the distance variable belonging at a gamma distribution. Fire model was configured taking into account the Eq. (12).

To compare the three models, we focus on the provider agent 3 with an 'ie' value, the real quality service, equal to 8.13 (extracted from Table 6). We can see in Table 8 the trust average values and the percentages of error calculated with this ie for the three reputation systems. The comparison between CALoR and Fire taking into account the provider 3 is shown in

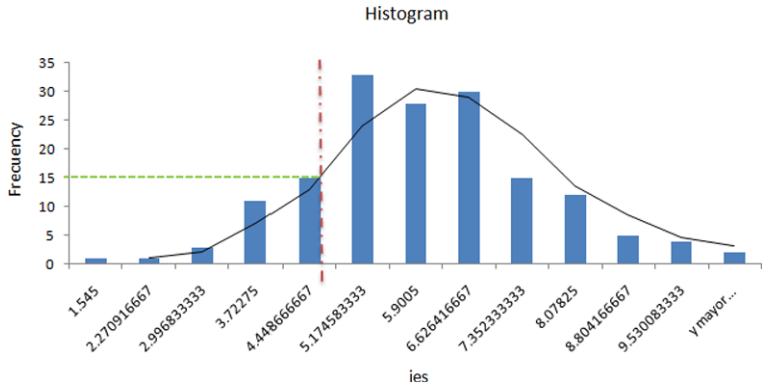

Fig. 11. Histogram of ies of PA2 and its final reputation considering distance.

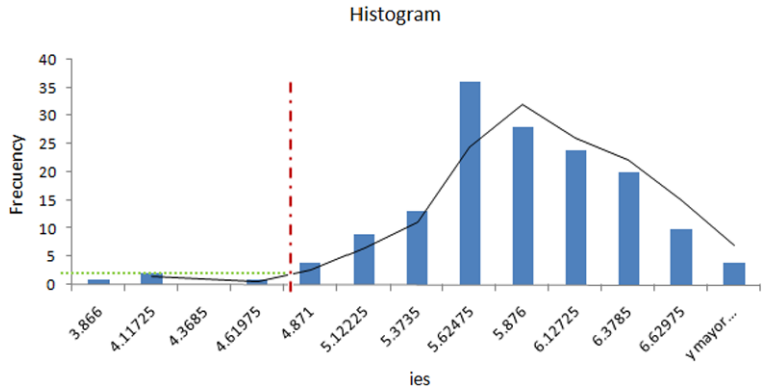

Fig. 12. Histogram of ies of PA2 and its final reputation ignoring distance.

Table 8
Trust values average and errors of PA3

	Fire	CALoR	Bishr et. all
Average	8.07755102	8.17071429	6.926632653
Error	0.6451289	−0.50079072	14.80156638

Fig. 13, and the comparison between CALoR and the propose of Bishr is shown in Fig. 14. From them we can estimate that CALoR system seems to converge more quickly than the other systems and it seems to present more stability. Nevertheless a deeper comparison should be performed to confirm this statement.

$$T_k = \frac{w_k \cdot T_i}{w_k} \tag{12}$$

Where the weight w_k is the freshness function and T_i is the trust value calculated based-on social reputation. The configuration of the experiment was: T_i a data from a normal distribution in a range [1,10]; w_k a value from a gamma distribution, in a range of [0,1].

The model proposed by Bishr present an error of the 14.8% while the error of Fire is the 0.64 and the error calculated with CALoR system is the -0.5. Since the differences in average values of trust are small (8.07.. vs 8.17..), other comparison criteria (such as the velocity of trust convergence to very close evaluations) could also be considered.

8. Conclusions

We designed a reputation system for Ambient Intelligence domains and dubbed it: Context-Aware and Location Reputation System: CALOR. The main contribution of this paper is the inclusion of geo-spatial information relative to interactions jointly with how much recent was the interaction (their *freshness*). In order to achieve this goal, we defined the required new concepts to support reputation communication and computations in an ontology for AmI domains, proposed previously by us [9]. This extension of the Protégé ontology includes a public profile of the user

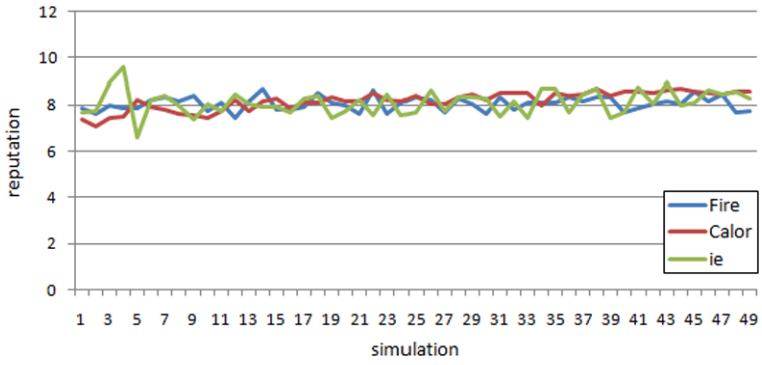

Fig. 13. Performance of CALoR and Fire evaluating PA3' behavior.

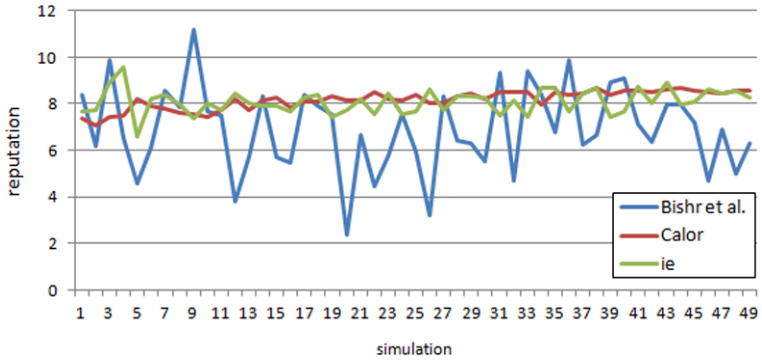

Fig. 14. Performance of CALoR and Bishr evaluating PA3' behavior.

agent obtained in a non-intrusive way through external observation of agents movements, and it contains the two relevant structured concepts to be generated and exchanged in the recommendation protocol: interactions fulfill and recommendation received.

We have suggested a specific formulation of such inclusion of temporal and spatial references considering previous works of state of the art. We have also validated our reputation system using an illustrative use-case. Furthermore, we have implemented the agent system on a JADE platform. And, finally, we have run simulations to show the advantage of our system when distance considerations are introduced in Ambient Intelligence domain problems. Further works will involve the evaluation of different initial setups representing extreme situations of Ambient Intelligence with our CALoR model.

Acknowledgements

This work was supported in part by Projects CICYT TIN2011-28620-C02-01, CICYT TEC2011-28626-C02-02, CAM CONTEXTS (S2009/TIC-1485), DPS2008-07029-C02-02 and MINECO TEC2012-37832-C02-01.

References

[1] F.L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology, Wiley, April 2007.

[2] C. Bertocco, Context-dependent reputation management in multi-agent systems, PhD thesis, Universitá de Padova, 2009.

[3] M. Bishr and L. Mantelas, A trust and reputation model for filtering and classifying knowledge about urban growth, *GeoJournal* (July 2008), 229–237.

[4] T. Garcia-Valverde, E. Serrano, and J.A. Botia, *Combining the Real World with Simulations for a Robust Testing of Ambient Intelligence Services*, Springer, 2012.

[5] T.D. Hunyh, Trust and reputation in open multi-agent systems (FIRE), PhD thesis, University of Southampton, 2006.

[6] A. Josang, R. Ismail, and C. Boy, A survey of trust and reputation systems for online service provision, *Decision Support Systems* (July 2006), 618–644.

[7] L. Navarro, F. Flacher, and V. Corruble, Dynamic level of detail for large scale agent-based urban simulations, in: *AAMAS*, 2011, pp. 701–708.

[8] J. Tsai, N. Fridman, E. Bowring, M. Brown, S. Epstein, G.A. Kaminka, S. Marsella, A. Ogden, I. Rika, A. Sheel,

M.E. Taylor, X. Wang, A. Zilka, and M. Tambe, Escapes: Evacuation simulation with children, authorities, parents, emotions, and social comparison, in: *AAMAS*, 2011, pp. 457–464.

[9] V. Venturini, J. Carbo, and J.M. Molina, An ambient intelligent platform based on multi-agent system, in: *XI Workshop of Physical Agents*, September 2010.

[10] V. Venturini, J. Carbo, and J.M. Molina, Methodological design and comparative evaluation of a mas providing ami, *Expert Systems With Applications Journal* (March 2012).

[11] D. Wright, S. Gutwirth, M. Friedewald, E. Vildjiounaite, and Y. Punie, *Safeguards in a World of AmI*, Springer, 2008.